SWIF and hdswif

August 11, 2015
Kei Moriya

Last updated 2015/10/07

ARIZONA STATE UNIVERSITY

SWIF

- Scientific Workflow Indefatigable Factotum
- Developed by Chris Larrieu in SciComp group larrieu@jlab.org
- https://scicomp.jlab.org/docs/swif
- Advantages:
  - Better control
    - Auger does not relate tape retrieval and jobs efficiently
    - Auger submission is slow (Java parsing XML)
    - Control over jobs in phases
  - Info on jobs
    - Success/problems
    - Resource usage

ASU Arizona State University

Using SWIF Directly

```
> swif create my_workflow
```

create workflow

swif command

swif options

```
> swif add-job -workflow my_workflow -project gluex -track reconstruction -cores 6 -disk 40g
-ram 5g -time 8h -os centos65
-input hd_rawdata_002333_000.evio mss:/mss/halld/RunPeriod-2014-10/rawdata/Run002333/
hd_rawdata_002333_000.evio
-stdout /home/gxproj1/logfiles/stdout
-stderr /home/gxproj1/logfiles/stderr
-name my_job
/home/gxproj1/script.sh TAGH_online 002333 000 6
```

script to execute with options

register job

```
> swif run my_workflow
```

run jobs: SWIF will dispatch jobs as it deems fit

```
> swif status my_workflow
```

check status

```
> swif status my_workflow -runs
```

check status for individual jobs

```
> swif cancel my_workflow -delete
```

cancel jobs, option -delete removes workflow

hdswif

- For most part just a wrapper around swif, but also
  - use config file to set up jobs more easily
  - organize SWIF output to check job status
- Check out from svn: https://halldsvn.jlab.org/repos/trunk/scripts/monitoring/hdswif
- Written in python, uses PyROOT (officially ROOT supported Python extention) for ROOT functionality (histograms, graphics, etc.)
- Offline monitoring code developed by Sean was heavily referenced for development, config file similar to Nathan's Julia scripts

ASU Arizona State University

How swif works

- Can run by typing swif in ifarm terminal
- Jobs are grouped into workflows, most natural to have separate workflows for different kinds of jobs
- Can get info with

```
> swif help
> swif help [command]
```

- List all workflows:

```
> swif list
```

- Show workflow status:

```
> swif status [workflow]
```

ASU Arizona State University

Setting Up hdswif

- Check out source code: https://halldsvn.jlab.org/repos/trunk/scripts/monitoring/hdswif
- Provide valid $ROOTSYS compiled with PyROOT enabled, have $ROOTSYS/bin in PATH, $ROOTSYS/lib in LD_LIBRARY_PATH
- hdswif will set PYTHONPATH to contain $ROOTSYS/bin internally
- The officially supported ROOT builds on the JLab ifarm CentOS6.5

  /group/halld/Software/builds/Linux_CentOS6-x86_64-gcc4.4.7/root/prod

  has PyROOT enabled

ASU Arizona State University

Wrappers in hdswif

- All of the following are equivalent:

swif command	hdswif command	what it does
swif list	hdswif.py list	Lists all workflows for user
swif create [workflow]	hdswif.py create [workflow]	Create new workflow
swif status [workflow]	hdswif.py status [workflow]	Show status of workflow
swif run [workflow]	hdswif.py run [workflow]	Start submitting jobs
swif cancel [workflow]	hdswif.py cancel [workflow]	Cancel running jobs
swif freeze [workflow]	hdswif.py freeze [workflow]	Disable deletion of workflow
swif unfreeze [workflow]	hdswif.py unfreeze [workflow]	Allow deletion of workflow

Commands wrapped for convenience, but calling swif directly is faster

Additions in hdswif.py

hdswif.py command	result
hdswif.py fullstatus [workflow] [format]	Show status of all runs, equivalent to swif status [workflow] -runs -summary -display [format] [format] is one of simple, xml, json
hdswif.py run [workflow] [njobs]	Run only [njobs], equivalent to swif run [workflow] -joblimit
hdswif.py delete [workflow]	Stop all running jobs and delete workflow. Equivalent to swif cancel [workflow] -delete
hdswif.py summary [workflow]	Create XML file for all jobs, and summary webpage of results based on this XML file

Adding Jobs

- In swif:

swif command

swif options

```
> swif add-job -workflow my_workflow -project gluex -track reconstruction -cores 6 -disk 40g
-ram 5g -time 8h -os centos65
-input hd_rawdata_002333_000.evio mss:/mss/halld/RunPeriod-2014-10/rawdata/Run002333/
hd_rawdata_002333_000.evio
-stdout /home/gxproj1/logfiles/stdout
-stderr /home/gxproj1/logfiles/stderr
-name my_job
/home/gxproj1/script.sh TAGH_online 002333 000 6
```

script to execute with options

- In hdswif:

```
> hdswif.py add my_workflow -c myconfig.txt
```

all options handled with config file

Config File Example

- Included example input.config:

```
PROJECT                         gluex
TRACK                           reconstruction
OS                              centos65

NCORES                          2
DISK                            40
RAM                             5
TIMELIMIT                       8

JOBNAMEBASE                     myjobs
RUNPERIOD                       2015-03
VERSION                         99
OUTPUT_TOPDIR                   /volatile/halld/test/RunPeriod-[RUNPERIOD]/ver[VERSION]
SCRIPTFILE                      /home/gxproj5/halld/hdswif/script.sh
ENVFILE                         /home/gxproj5/halld/hdswif/setup_jlab-2015-03.csh
```

Example of other variables included in variable.

Must specify full path

- These will overwrite defaults - safer to specify them in config file

Config File Variables

- Variables should be obvious for the most part
- RUNPERIOD is used to look for file in /mss/halld/RunPeriod-[RUNPERIOD]/rawdata/ for evio data files
- VERSION is a variable for offline monitoring - do not need to use
- Config variables can depend on each other:

```
OUTPUT_TOPDIR       /volatile/halld/test/RunPeriod-[RUNPERIOD]/ver[VERSION]
```

- Replacements will be made (do not put in cyclic dependencies)

User Tags

- To easily access information about jobs, SWIF allows user-defined tags
- Specify with -tag [tag name] [tag value] when using SWIF
- hdswif gives each job tags user_run and user_file for the run and file specified for parsing output

Specifying Runs/Files

- Specify runs, files with -r [RUN] and -f [FILE]
- If option -r is not specified, will run over ALL runs in RUNPERIOD
- Same for files
- Can use UNIX-style wildcards

```
> hdswif.py add my_workflow
```

Run over all runs, files in RUNPERIOD

```
> hdswif.py add my_workflow -r 3185
```

Run over all files in run 003185

```
> hdswif.py add my_workflow -r 3185 -f 1
```

Run over file 001 in run 003185

```
> hdswif.py add my_workflow -r '318[05]' -f '01[0-9]'
```

Run over files 010-019 in run 003180, 003185

ASU Arizona State University

Re-Submitting Jobs

- In swif:

swif command

modification

problem type

```
> swif modify-jobs -workflow my_workflow -ram add 2gb -problems AUGER-OVER_RLIMIT
```

- In hdswif:

problem

Additional option for adding memory (+2GB is default)

```
> hdswif.py resubmit my_workflow TIMEOUT 3
```

Currently problem can be:

- SYSTEM (resubmit all jobs with AUGER-FAILED, AUGER-INPUT-FAIL, AUGER-OUTPUT-FAIL, SWIF-SYSTEM-ERROR with same resources)
- RLIMIT (default adds 2GB RAM)
- TIMEOUT (default adds 2hrs)

Job Summary

- In hdswif:

```
> hdswif.py summary my_workflow
```

- Creates html page summary_swif_output_[workflow].html

Note:
Need CSS file mystyle.css for html formatting, included in svn

summary of jobs

offline_monitoring_RunPeriod2015_03_ver11_hd_rawdata

Status

Info retrieved: 2015-08-10 07:30:19.0

Job Limit: 0

Total attempts: 4314

Undispatched	Dispatched	Succeeded	Failed	Problems	Canceled	Total
0	0	4237	0	77	0	**4314**

results

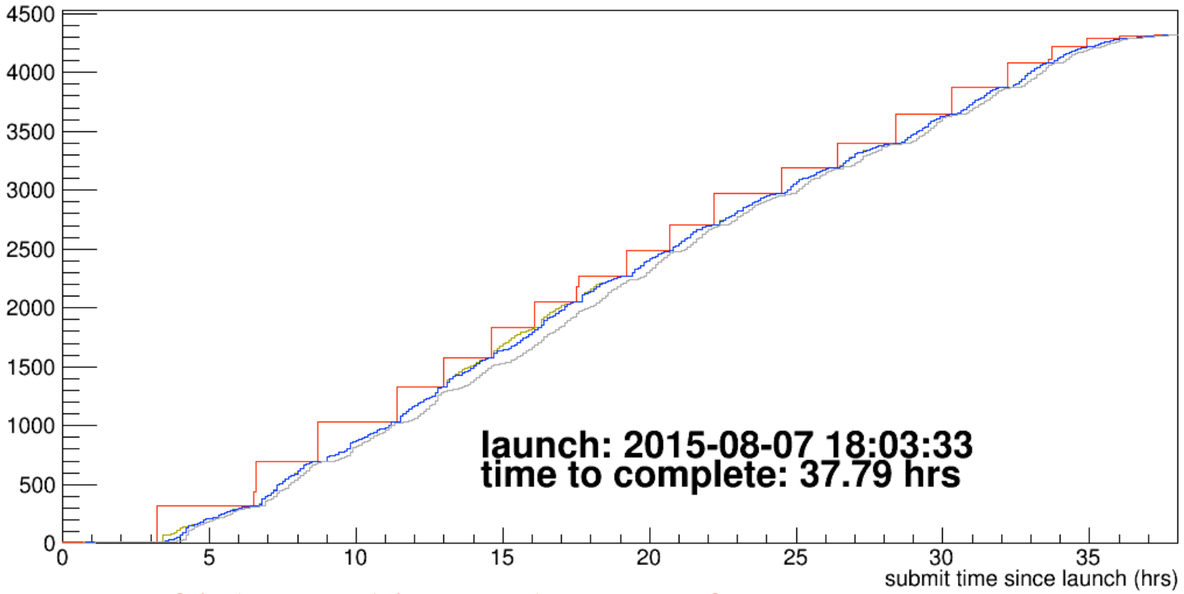
Auger Status

Dependency	Pending	Staging In	Active	Staging Out	Finishing	Dispatched Total
0	0	0	0	0	0	**0**

active jobs

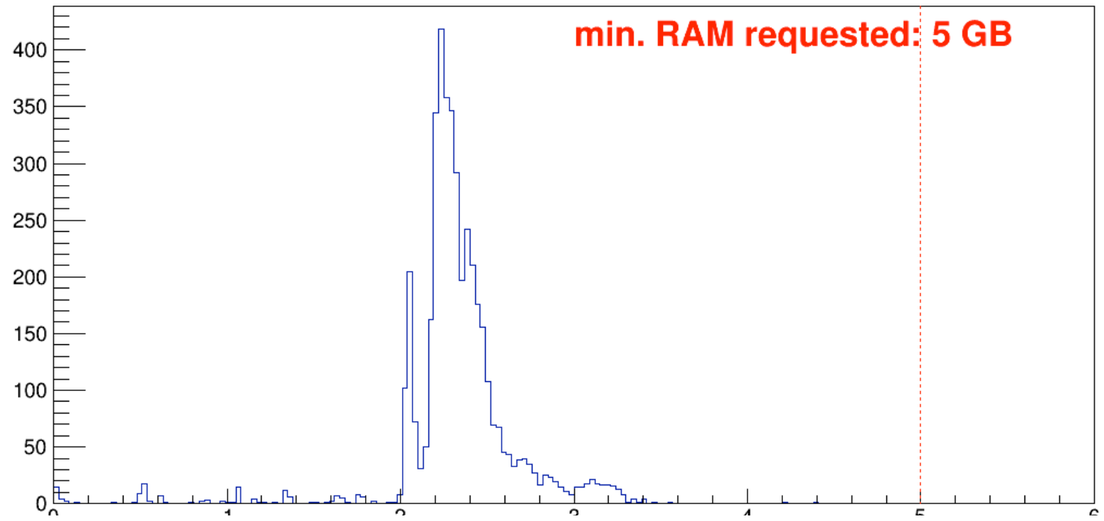
Job Summary Figures

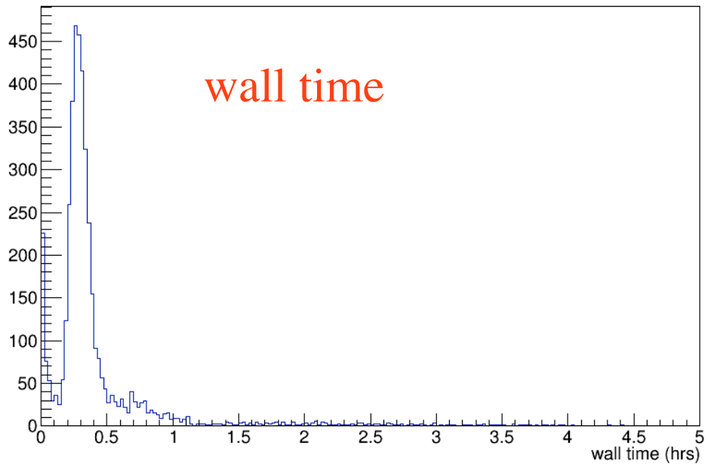
- All included in same generated webpage

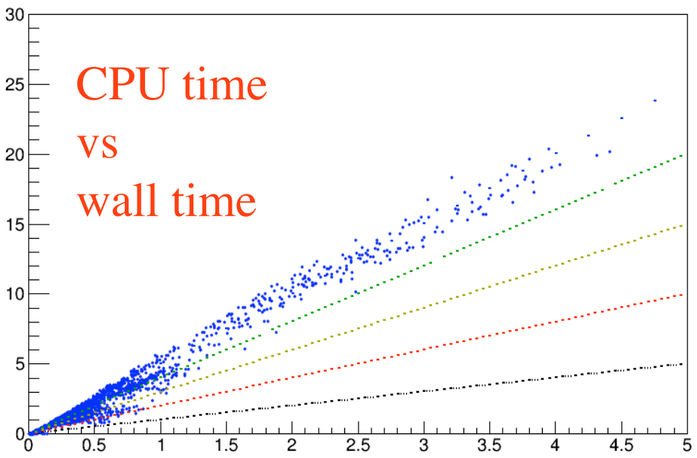


# of jobs reaching each stage of processing against time

RAM usage







Job Details

- Even if you forget details of registered jobs(for example by editing the script file for that job) you can retrieve what is stored in SWIF
- The command is `curl http://farmpbs:6080/job/[job id]/spec`
  where the [job id] is a unique number given to each job by SWIF.

- The wrapper command within hdswif is `hdswif.py details [workflow] -r [run] -f [file]`
  where it is **assumed** that the user specified the user tags `user_run` and `user_file` (this is the default if hdswif was used to register the job)

```
ifarm1401> hdswif.py details offline_monitoring_RunPeriod2015_03_ver16_hd_rawdata -r 3185 -f 000
Creating XML output file........
--- Found match #1 ---
name = offmon__003185_000
project = gluex
track = reconstruction
os = centos65
cpu_cores = 6
disk_bytes = 40000000000
ram_bytes = 8000000000
time_secs = 57600
script = /home/gxproj5/halld/hdswif/script.sh /home/gxproj5/halld/hdswif/setup_jlab-2015-03.csh hd_rawdata_003185_000.evio 003185 000 /volatile/halld/
offline_monitoring/RunPeriod-2015-03/ver16/ 6
local_uri = hd_rawdata_003185_000.evio
remote_uri = /mss/halld/RunPeriod-2015-03/rawdata/Run003185/hd_rawdata_003185_000.evio
=======================================
Total of 1 jobs found that matched
```

example output

hdswif Cheatsheet

hdswif command	what it does
hdswif.py create [workflow]	Create workflow
hdswif.py status [workflow] (format)	Check status
hdswif.py add [workflow] -c (config file) -r (runs) -f (files)	Add jobs to workflow
hdswif.py run [workflow]	Run workflow
hdswif.py run [workflow] (n)	Only run n jobs
hdswif.py resubmit [workflow] (n)	Resubmit jobs with nGB of additional RAM
hdswif.py cancel [workflow]	Cancel all dispatched jobs
hdswif.py delete [workflow]	Delete workflow

ASU Arizona State University

TO DO

- Working with Chris on quicker batch submission
- Deletion of registered jobs
- Resubmission of succeeded jobs

- Questions, comments, requests are very welcome